

**Ejercicio 1(20)** Realice la gramática EBNF para la definicion de un registro de pascal, ejemplo:

record nombre of

campo1:tipo

campo2:tipo

....

end

b) (5) Realice el diagrama sintáctico del ejercicio anterior.

**Ejercicio 2** Sea el siguiente código en Ada, indique para todos los identificadores indicando el número de línea

a) (5) Su tipo de ligadura con l-valor.

b) (5) Su r-valor al momento de declaración (inicialización).

c) (5) Alcance y d) (5) tiempo de vida.

SE ENMASCARA

4-618-13

```

1.  with text_io; use text_io;
2.  Procedure Principal is;
3.  type puntero is access integer;
4.  e, k, l:integer;
5.  m: constant integer:=8;
6.  Procedure Uno is;
    -> 6.1.  k:char
    -> 6.2.  l: puntero;
        6.3.  begin
            6.3.1.  k:="Z";
            6.3.2.  k:="X";
            6.3.3.  l:= new puntero;
            6.3.4.  ..
            6.3.5.  free l;
        6.4.  end;
7.  Procedure Dos is;
    7.1.  k:boolean
    7.2.  e:char;
    7.3.  begin
        7.3.1.  k:=T.;
        7.3.2.  e:= "X";
        7.3.3.  l:=m;
        7.3.4.  l:=7;
    7.4.  end;
8.  begin
9.  k:=5;
10. Uno;
11. Dos;
12. e:= k + 2;
13. end.

```

Realice este ejercicio sobre esta misma hoja.

Identif	L-value	R-Value	Alcance	T.V.
PRINCIPAL	-	-	2-13	2-13
E(4)	AUTOM	BASERA	4-7.1-8-13	2-13
K(4)	AUTOM	BASERA	-	2-13
L(4)	AUTOM	BASERA	4-6.1-7.2	2-13
M(5)	AUTOM	BASERA	5-13	2-13
UNO(6)	-	-	6-13	6-6.4
K(6.1)	AUTOM	BASERA	6.1-6.4	6-6.4
L(6.2)	AUTOM	NIL	6.2-6.4	6-6.4
L(6.2)	DINAM	BASERA	6.2-6.4	6.3-6.5
DOS(7)	-	-	7-13	7-7.4
K(7.1)	AUTOM	BASERA	7.1-7.4	7-7.4
E(7.2)	AUTOM	BASERA	7.2-7.4	7-7.4

**Ejercicio 4** Sea el siguiente programa escrito en Pascal-like, realice la pila de ejecución,  
**a) (15)** Por cadena estática **b) (15)** Por cadena dinámica

```
Program Main;
  Const j:= 3;
  Type arre= array [1..6] of integer;
  Var m, n, z:integer; a, b: arre;
  Procedure B(ref m:integer, val-res p:integer);
    var n,c: integer;

    begin
      n:= m + 3; p:=8; c:=3;
      D; m:= m + 1;
      a(n-3):=10;
    end
  Procedure D;
    begin
      m:= c + n +m; m:= m +1; n:= n + 2;
    end
  Function C: integer;
    begin
      b(m):= b(m) + 1; m:= m + 1;
      return b(m);
    end
  begin
    m:= 1; n:= 2;
    for z:=1 to 6 do begin
      a(z):= z; b(z):= z + 2;
    end;

    B(n,z);
    for z:=1 to 6 do write (a(z), b(z));
  end.
```

Nota: La forma de evaluación del lenguaje es de izquierda a derecha

**Ejercicio 5 a) (10)** Dado el siguiente código en Java. Describa los posibles caminos de ejecución.

```
public static void main(String[] args) {
  Double array_doubles[] = new Double[10];
  for (int i = 0; i < array_doubles.length; i++) {
    array_doubles[i] = 5.00 * i;
  }
  for (int i = 0; i < array_doubles.length; i++) {
    try {
      .....
      if ((i % 2) == 0) {
        array_doubles[i] = 5 * valorModificado(array_doubles, i);
      }
      .....
    } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println(e.toString());
    }
  }
}
```

```

        } catch (Exception a) {
            System.out.println(a.toString());
        } finally {
            System.out.println("proceso Terminado");
        }
    }
    .....
    .....
}

public static double valorModificado(Double[] v, int x) throws Exception, ArrayIndexOutOfBoundsException {
    if ((x >= 0) && (x < 8)) {
        return (x * 2);
    } else {
        if (x < 0) {
            // caso excepcional
            throw new ArrayIndexOutOfBoundsException(" el valor " + x
                + " es un número negativo");
        } else {
            // caso excepcional
            throw new Exception(" el valor " + x + " no es menor que 8");
        }
    }
}
}

```

**b)(10p).** En el siguiente código de PL1, grafique cómo se apilan los manejadores. y el valor que imprime en el final de la ejecución.

```

Program Nuevo;
var x:int; b1,b2:boolean;
Procedure Pri (b:boolean);
    var x:int;
    begin
        On condition UNO begin x:=x+25 end;
        x:=1;
        if b=true then SignalCondition UNO;
        x:=x+4;
        If (not b)=true then Signal Condition DOS
    end;
Begin
    x:=4;
    On condition UNO begin x:=x+1 end;
    b2:=true;
    b1:=false;
    On condition DOS begin x:=x*100end;
    if b1=false then SignalCondition DOS;
    Pri(b1);
    write (x);
End.

```